Harvesting and Storing Knowledge from the Web

Eduard Hovy
Information Sciences Institute
University of Southern California
The problem with automated QA...

• Where do lobsters like to live?  
  — on the table

• Where are zebras most likely found?  
  — in the dictionary

• How many people live in Chile?  
  — nine

• What is an invertebrate?  
  — Dukakis

BUT: we could quite easily get this from the web!  
…need a repository of knowledge plus commonsense semantic, numerical info
Imagine…

The web is the world’s knowledge store.
But it is disorganized, inconsistent, and constantly changing…
…and it’s often hard to find things quickly and accurately.

WHAT IF you could ask your system to create a database of the knowledge you needed, overnight?

You’d need, at least:

- metadata creation
- query input / definition
- data item harvesting

- data relationship harvesting
- data verification
- data updating
Example applications 1: NLP

• **How to improve accuracy of IR / web search?**
  TREC 98–01: around 40%
  ★ Understand user query; expand query terms by meaning

• **How to achieve conceptual summarization?**
  Never been done yet, at non-toy level
  ★ Interpret topic, fuse concepts according to meaning; re-generate

• **How to improve QA?**
  TREC 99–02: around 65%
  ★ Understand Q and A; match their meanings; know common info

• **How to improve MT quality?**
  MTEval 94: ~70%, depending on what you measure
  ★ Disambiguate word senses to find correct meaning
Large standardized metadata collections

What is an ontology?
My def: a collection of terms denoting entities, events, and relationships in the domain, taxonomized and interrelated so as to express the sharing of properties. It’s a formalized model of the domain, focusing on the aspects of interest for computation.

The need is there…
everybody’s making lists:
  • SIC and NAICS and other codes
  • Yahoo!’s topic classification
  • Semantic Web termbanks / ontologies

But how do you:
  • Guarantee the freshness and accuracy of the list?
  • Guarantee its completeness?
  • Ensure commensurate detail in levels of the list?
  • Cross-reference elements of the list?

Need automated procedure for creating lists / metadata / ontologies
Credo and methodology

Ontologies (and even concepts) are too complex to build all in one step…

…so build them bit by bit, testing each new (kind of) addition empirically…

…and develop appropriate learning techniques for each bit, so you can automate the process…

…so next time (since there’s no ultimate truth) you can build a new one more quickly
Plan: stepwise accretion of knowledge

• Initial framework:
  – Start with existing (terminological) ontologies as pre-metadata
  – Weave them together

• Build metadata/concepts:
  – Define/extract concept ‘cores’
  – Extract/learn inter-concept relationships
  – Extract/learn definitional and other info

• Build (large) data/instance base:
  – Extract instance ‘cores’
  – Link into ontology; store in databases
  – Extract more information, guided by parent concept
Talk overview

1. **Framework**: Ontology as metadata
   - Creating Omega: recent work on connecting ontologies

2. **Metadata terms (concepts)**:
   - Learning concepts by clustering
   - Learning additional relations from text

3. **Data (instances)**:
   - Harvesting seed instances from text
   - Harvesting additional information

4. **Verifying data**:
   - Determining opinions and other epistemic statuses
   - Updating

5. **Conclusion**
1. Framework

Ontology as metadata: semi-automated alignment and merging

(This work with Andrew Philpot, Michael Fleischman, and Jerry Hobbs)
Example application: EDC  
(Hovy et al. 02)
Our own new work (ISI) 400 nodes

Penman Upper Model (ISI) 300 nodes

Mikrokosmos (New Mexico State U) 6,000 nodes

Omega (Hovy et al. 03)

WordNet 1.7 (Princeton) 110,000 nodes
General alignment and merging problem

Goal: **find attachment point(s) in ontology** for node/term from somewhere else (ontology, website, metadata schema, etc.)

It’s hard to do manually; very hard to do automatically — system needs to understand semantics of entities to be aligned

**Existing Ontology 1**

**Alignment algorithms**

**Acceptance procedure**

**Reference Ontology**

**Existing Ontology 2**

**Merge Structure**

**Validation in context**
Ontology alignment and merging

• Goal: find attachment point in ontology for node/term from somewhere else (ontology, website, metadata schema, etc.)
• Procedure:
  – 1. For a new term/concept, extract and format: name, definition, associated text, local taxonomy cluster, etc.
  – 2. apply alignment suggestion heuristics (NAME, DEFINITION, HIERARCHY, DISPERSAL match) against big ontology, to get proposed attachment points with strengths (Hovy 98) — test with numerous parameter combinations, see http://edc.isi.edu/alignment/ (Hovy et al. 01)
  – 3. automatically combine proposals (Fleischman et al 03)
  – 4. apply verification checks
  – 5. bless or reject proposals manually
• Process developed in early 1990s: (Agirre et al. 94; Knight & Luk 94; Okumura & Hovy 96; Hovy 98; Hovy et al. 01)
• Not stunningly accurate, but can speed up manual alignment markedly
Alignment for Ωmega

- Created Upper Region (400 nodes) manually
- Manually snipped tops off Mikro and WordNet, then attached them to fringe of Upper Region
- Automatically aligned bottom fringe of Mikro into WordNet
- Automatically aligned sides of bubbles
Dynamically generated setup-up page

For merge point U@U[VERTEBRATE], micro leafM25

<table>
<thead>
<tr>
<th>U@U[VERTEBRATE]</th>
<th>S@l-reanu....</th>
<th>S@b-laminal</th>
<th>S@p-lacant....</th>
<th>S@k-lheald</th>
<th>S@t-lshaped</th>
<th>S@l-labial-oral</th>
</tr>
</thead>
<tbody>
<tr>
<td>M25@I[S5A-MAMMAL]</td>
<td>S@l-reanu....</td>
<td>S@b-laminal</td>
<td>S@p-lacant....</td>
<td>S@k-lheald</td>
<td>S@t-lshaped</td>
<td>S@l-labial-oral</td>
</tr>
<tr>
<td>M25@I[DOLPHIN]</td>
<td>S@l-reanu....</td>
<td>S@b-laminal</td>
<td>S@p-lacant....</td>
<td>S@k-lheald</td>
<td>S@t-lshaped</td>
<td>S@l-labial-oral</td>
</tr>
<tr>
<td>M25@I[BOTTLENOSE]</td>
<td>S@l-reanu....</td>
<td>S@b-laminal</td>
<td>S@p-lacant....</td>
<td>S@k-lheald</td>
<td>S@t-lshaped</td>
<td>S@l-labial-oral</td>
</tr>
</tbody>
</table>

For 24 pairings, there were 6 possible matches:

1. [ACCEPT] M25@I[DOLPHIN] <-> S@l-dolphin
2. [ACCEPT] M25@I[S5A-MAMMAL] <-> S@l-
3. [ACCEPT] M25@I[S5A-MAMMAL] <-> S@l-
4. [ACCEPT] U@U[MAMMAL] <-> S@l-cestacse
5. [ACCEPT] U@U[MAMMAL] <-> S@l-placenta
6. [ACCEPT] U@U[MAMMAL] <-> S@l-mamnald

The maximal consistent subset size is 3. There were 3 matches:

1. [ACCEPT] 1. M25@I[DOLPHIN] <-> S@l-dolphin
2. M25@I[S5A-MAMMAL] <-> S@l-kel
3. U@U[MAMMAL] <-> S@l-mamnald

2. [ACCEPT] 1. M25@I[DOLPHIN] <-> S@l-dolphin
2. M25@I[S5A-MAMMAL] <-> S@l-kel
3. U@U[MAMMAL] <-> S@l-mamnald

3. [ACCEPT] 1. M25@I[DOLPHIN] <-> S@l-dolphin
2. M25@I[S5A-MAMMAL] <-> S@l-kel
3. U@U[MAMMAL] <-> S@l-mamnald

Next Page
A puzzle...

- Is Amber Decomposable or Nondecomposable?
- The ‘stone’ sense of it (Mikro) is; the ‘resin’ sense (WordNet) is not...
- What to do??

Dynamically generated sewing up page

For merge point U[@::TANGIBLE-OBJECT] (orig merge point was U[@::NATURAL-OBJECT]), mikro leaf M25@::JABER),

sensor interior concept U[@:amber<<resin]
Shishkebobs

(Hovy et al. 03)

• Library ISA Building (and hence can’t buy things)
  Library ISA Institution (and hence can buy things)
  SO: Building ◊ Institution ◊ Location …a Library is all these

• Also: Field-of-Study ◊ Activity ◊ Result-of-Process:
  (Science, Medicine, Architecture, Art…)

• Allowing shishkebobs makes merging ontologies easier
  (possible?): you respect each ontology’s perspective

• Continuum: from on-the-fly shadings to metonymy
  (see Guarino’s identity conditions; Pustejovsky’s qualia)

• We found about 400 shishkebobs
2a. New Metadata:
Learning terms by clustering web information

(This work by Patrick Pantel and Dekang Lin)
Where/how to find new metadata?

• Potential sources:
  – Existing ontologies (AI efforts, Yahoo!, etc.) and lists (SIC codes, etc.)
  – Manual entry, esp with reference to foreign-language text (EuroWordNet, IL-Annot, etc.)
  – Dictionaries and thesauri (Webster’s, Roget’s, etc.)
  – Automated discovery by text clustering (Pantel and Lin, etc.)

• Issues:
  – How large do you want it? — tradeoff size vs. consistency and ease of use
  – How detailed? — tradeoff granularity/domain-specificity vs. portability and wide acceptance (Semantic Web)
  – How language-independent? — tradeoff independence vs. utility for non/shallow-semantic NLP applications
Clustering By Committee (Pantel and Lin 02)

- Very accurate new clustering procedure CBC:
  - Define syntactic/POS patterns as features (N-N; N-subj-V)
  - Parse corpus using MINIPAR (D. Lin)
  - Cluster, using MI on features:
    \[
    mi_{ef} = \log \frac{c_{ef}}{N} \times \frac{\sum_{i=1}^{m} c_{eq}}{N}
    \]
    
  - Find cluster centroids (using strict criteria): committee
  - For non-centroid words, match their pattern features to committee’s; if match, include in cluster, remove features
  - If word has remaining features, try to include in other clusters as well — handle ambiguity

- Find name for clusters (superconcepts):
  - Word shared in apposition, nominal-subj, etc. templates

- Complexity: \(O(n^2k)\) for \(n\) words in corpus, \(k\) features
Abraham Lincoln, Thomas Jefferson, George Washington
(N221) - 0.198327

Porsche, Cadillac, Corvette
(N1247) - 0.154288

Encino, Woodland Hills, Sherman Oaks
(N774) - 0.133465
<table>
<thead>
<tr>
<th>N221: Abraham Lincoln, Thomas Jefferson, George Washington</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abraham Lincoln                                         0.440408</td>
</tr>
<tr>
<td>Thomas Jefferson                                        0.427359</td>
</tr>
<tr>
<td>George Washington                                       0.390025</td>
</tr>
<tr>
<td>Alexander Hamilton                                      0.357866</td>
</tr>
<tr>
<td>Franklin Delano Roosevelt                               0.347813</td>
</tr>
<tr>
<td>Andrew Jackson                                          0.346218</td>
</tr>
<tr>
<td>John Adams                                               0.333103</td>
</tr>
<tr>
<td>Theodore Roosevelt                                      0.254847</td>
</tr>
<tr>
<td>Jefferson                                                0.232393</td>
</tr>
<tr>
<td>Franklin Roosevelt                                      0.231732</td>
</tr>
<tr>
<td>John Quincy Adams                                        0.20884</td>
</tr>
<tr>
<td>Benjamin Franklin                                        0.20414</td>
</tr>
<tr>
<td>Lincoln                                                  0.198327</td>
</tr>
<tr>
<td>James Madison                                            0.197463</td>
</tr>
<tr>
<td>Winston Churchill                                        0.194458</td>
</tr>
<tr>
<td>Robert E. Lee                                            0.174996</td>
</tr>
<tr>
<td>Roosevelt                                                0.164226</td>
</tr>
<tr>
<td>founding father                                          0.161931</td>
</tr>
<tr>
<td>Martin Luther King Jr.                                   0.136376</td>
</tr>
<tr>
<td>Napoleon                                                 0.128418</td>
</tr>
<tr>
<td>Christopher Columbus                                     0.128006</td>
</tr>
<tr>
<td>Sam Houston                                              0.123654</td>
</tr>
<tr>
<td>Lenin                                                    0.123188</td>
</tr>
<tr>
<td>Mel Karmazin                                             0.120426</td>
</tr>
<tr>
<td>Genghis Khan                                             0.110591</td>
</tr>
</tbody>
</table>

Abraham Lincoln, Thomas Jefferson, George Washington (N221) - 0.198327

Porsche, Cadillac, Corvette (N1247) - 0.154288

Encino, Woodland Hills, Sherman Oakes (N774) - 0.133465
Grammatical templates

**-V:obj:N 1869 times:**

- {V1662 offer, provide, make} 156, have 108, {V1650 go, take, fly} 51, sell 45, {V1754 become, remain, seem} 34, ... give 24, {V1647 oppose, reject, support} 24, buy 21, {V1653 allocate, earmark, owe} 21, win 20 ...

**-N:conj:N 536 times:**

- {N719 Toyota, Nissan, BMW} 65, {N257 Cadillac, Buick, Lexus} 59, {N549 Philadelphia, Seattle, Chicago} 41, American Continental 20, Cadillacs 11, ...

**-V:by:N 50 times:**

- {V1662 offer, provide, make} 12, own 5, hire 4, target 4, write 3, buy 2, ...
Finding names for clusters

• Search for repeated appositions
  – “the President, Thomas Jefferson, …”
  – “Kobe Bryant, famous basketball star…”

• Check against ontology, if present

• Examples for Lincoln:
  – PRESIDENT(N891) - 0.187331
  – UNIT / BORROWER / THRIFT(N724) - 0.166958
  – CAR / DIVISION(N257) - 0.137333

  → Candidate metadata terms
2b. Metadata Concept (Term)
Relations:
Learning interconnectivity

(This work with Chin-Yew Lin, Mike Junk, Michael Fleischman, and Tom Murray)
Topic signature

Word family built around inter-word relations.

• **Def**: Head word (or concept), plus set of related words (or concepts), each with strength:
  \[ \{ T_k, (t_{k1}, w_{k1}), (t_{k2}, w_{k2}), \ldots, (t_{kn}, w_{kn}) \} \]

• **Problem**: Scriptal co-occurrence, etc. — how to find it?
• Approximate by simple textual term co-occurrence...

**Related words in texts show Poisson distribution:**
In large set of texts, topic keywords concentrate around topics; so compare topical word frequency distributions against global background counts.
Learning signatures

Procedure:

1. Collect texts, sorted by topic
2. Identify families of co-occurring words
3. Evaluate their purity
4. Find the words’ concepts in the Ontology
5. Link together the concept signatures

Need texts, sorted
How to count co-occurrence?
How to evaluate?
Need disambiguator
Calculating weights

\[
\begin{align*}
\text{tf.idf} & : w_{jk} = \text{tf}_{jk} \times \text{idf}_j \\
\chi^2 & : w_{jk} = \begin{cases} 
\frac{(\text{tf}_{jk} - m_{jk})^2}{m_{jk}} & \text{if } \text{tf}_{jk} > m_{jk} \\
0 & \text{otherwise}
\end{cases}
\end{align*}
\]

(Hovy & Lin, 1997)

- \( \text{tf}_{jk} \): count of term \( j \) in text \( k \) ("waiter" often only in some texts).
- \( \text{idf}_j = \log(N/n_j) \): within-collection frequency ("the" often in all texts), 
  \( n_j \) = number of docs with term \( j \), \( N \) = total number of documents.
- \( \text{tf.idf} \) is the best for IR, among 287 methods (Salton & Buckley, 1988).
- \( m_{jk} = (\sum_j \text{tf}_{jk} \sum_k \text{tf}_{jk}) / \sum_j \text{tf}_{jk} \) : mean count for term \( j \) in text \( k \).

Likelihood ratio \( \lambda \):

\[
2 \log \lambda = 2N \cdot I(R;T)
\]

(Lin & Hovy, 2000)

(more approp. for sparse data; \(-2\log \lambda \) asymptotic to \( \chi^2 \)).

- \( N \) = total number terms in corpus.
- \( I \) = mutual information between text relevance \( R \) and given term \( T \),
  \( = H(R) - H(R \mid T) \) for \( H(R) \) = entropy of terms over relevant texts \( R \)
  and \( H(R \mid T) \) = entropy of term \( T \) over rel and nonrel texts.
Early signature study (Hovy & Lin 97)

• Corpus
  – Training set WSJ 1987:
    • 16,137 texts (32 topics).
  – Test set WSJ 1988:
    • 12,906 texts (31 topics).
  – Texts indexed into categories by humans.

• Signature data
  – 300 terms each, using \( tf.idf \).
  – Word forms: single words, demorphed words, multi-word phrases.

• Topic distinctness...
  – Topic hierarchy.

<table>
<thead>
<tr>
<th>RANK</th>
<th>ARO</th>
<th>BNK</th>
<th>ENV</th>
<th>TEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>contract</td>
<td>bank</td>
<td>epa</td>
<td>at&amp;t</td>
</tr>
<tr>
<td>2</td>
<td>air_force</td>
<td>thrift</td>
<td>waste</td>
<td>network</td>
</tr>
<tr>
<td>3</td>
<td>aircraft</td>
<td>banking</td>
<td>environmental</td>
<td>fcc</td>
</tr>
<tr>
<td>4</td>
<td>navy</td>
<td>loan</td>
<td>water</td>
<td>cbs</td>
</tr>
<tr>
<td>5</td>
<td>army</td>
<td>mr.</td>
<td>ozone</td>
<td>cable</td>
</tr>
<tr>
<td>6</td>
<td>space</td>
<td>deposit</td>
<td>state</td>
<td>bell</td>
</tr>
<tr>
<td>7</td>
<td>missile</td>
<td>board</td>
<td>incinerator</td>
<td>long-distance</td>
</tr>
<tr>
<td>8</td>
<td>equipment</td>
<td>fslic</td>
<td>agency</td>
<td>telephone</td>
</tr>
<tr>
<td>9</td>
<td>mcdonnell</td>
<td>fed</td>
<td>clean</td>
<td>telecomm.</td>
</tr>
<tr>
<td>10</td>
<td>northrop</td>
<td>institution</td>
<td>landfill</td>
<td>mci</td>
</tr>
<tr>
<td>11</td>
<td>nasa</td>
<td>federal</td>
<td>hazardous</td>
<td>mr.</td>
</tr>
<tr>
<td>12</td>
<td>pentagon</td>
<td>fdic</td>
<td>acid_rain</td>
<td>doctrine</td>
</tr>
<tr>
<td>13</td>
<td>defense</td>
<td>volcker</td>
<td>standard</td>
<td>service</td>
</tr>
<tr>
<td>14</td>
<td>receive</td>
<td>henkel</td>
<td>federal</td>
<td>news</td>
</tr>
</tbody>
</table>

\[
\text{ENV} \quad \text{TEL} \quad \text{FIN} \\
\quad \text{BNK} \quad \text{STK}
\]
Evaluating signatures

- **Solution**: Perform text categorization task:
  - create $N$ sets of texts, one per topic,
  - create $N$ topic signatures $TS_k$,
  - for each new document, create document signature $DS_i$,
  - compare $DS_i$ against all $TS_k$; assign document to best match.

- **Match function**: vector space similarity measure:
  - Cosine similarity, \( \cos \theta = \frac{TS_k \cdot DS_i}{\|TS_k\| \|DS_i\|} \).

- **Test 1** (Hovy & Lin, 1997, 1999)
  - Training set: 10 topics; $\sim$3,000 texts (TREC).
  - Contrast set (background): $\sim$3,000 texts.
  - Conclusion: $tf.idf$ and $\chi^2$ signatures work ok but depend on signature length.

- **Test 2** (Lin & Hovy, 2000):
  - 4 topics; 6,194 texts; uni/bi/trigram signats.
  - Evaluated using SUMMARIST: $\lambda > tf.idf$. 

![Average Recall and Precision Trend of Test Set WSJ7 PH](image-url)
Text pollution on the web

Goal: Create word families (signatures) for each concept in the Ontology. Get texts from Web.

Main problem: text pollution. What’s the search term?

Purifying: In later work, used Latent Semantic Analysis
Purifying with Latent Semantic Analysis

- Technique used in Psychologists to determine basic cognitive conceptual primitives (Deerwester et al., 1990; Landauer et al., 1998).
- Singular Value Decomposition (SVD) used for text categorization, lexical priming, language learning…
- LSA automatically creates collections of items that are correlated or anti-correlated, with strengths:

  *ice cream, drowning, sandals ⇒ summer*

- Each such collection is a ‘semantic primitive’ in terms of which objects in the world are understood.

- We tried LSA to find most reliable signatures in a collection—reduce number of signatures in contrast set.
LSA for signatures

• Create matrix $A$, one signature per column (words $\times$ topics).
• Apply SVDPAC to compute $U$ so that $A = U \Sigma U^T$:
  - $U$: $m \times n$ orthonormal matrix of left singular vectors that span space
  - $U^T$: $n \times n$ orthonormal matrix of right singular vectors
  - $\Sigma$: diagonal matrix with exactly $\text{rank}(A)$ nonzero singular values; $\sigma_1 > \sigma_2 > \ldots > \sigma_n$

• Use only the first $k$ of the new concepts: $\Sigma' = \{\sigma_1, \sigma_2 \ldots \sigma_k\}$.
• Create matrix $A'$ out of these $k$ vectors: $A' = U \Sigma' U^T \approx A$.

$A'$ is a new (words $\times$ topics) matrix, with different weights and new ‘topics’. Each column is a purified signature.
Some results with LSA  (Hovy and Junk 99)

- **Contrast set** (for *idf* and $\chi^2$): set of documents on very different topic, for good *idf*.
- **Partitions**: collect documents within each topic set into partitions, for faster processing. /n is a collecting parameter.
- **U function**: function for creation of LSA matrix.

## Results:
- Demorphing helps.
- $\chi^2$ better than *tf* and *tf.idf*.
- LSA improves results, but not dramatically.

### TREC texts

<table>
<thead>
<tr>
<th>Function</th>
<th>Demorph?</th>
<th>Partitions</th>
<th>U function</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Without contrast set</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>tf</em></td>
<td>no</td>
<td></td>
<td></td>
<td>0.748447</td>
<td>0.628782</td>
</tr>
<tr>
<td><em>tf</em></td>
<td>yes</td>
<td></td>
<td><em>tf</em></td>
<td>0.766428</td>
<td>0.737976</td>
</tr>
<tr>
<td><em>tf</em></td>
<td>yes</td>
<td>10</td>
<td><em>tf</em></td>
<td>0.820609</td>
<td>0.880663</td>
</tr>
<tr>
<td><em>tf</em></td>
<td>yes</td>
<td>20</td>
<td><em>tf</em></td>
<td>0.824180</td>
<td>0.882533</td>
</tr>
<tr>
<td><em>tf</em></td>
<td>yes</td>
<td>30</td>
<td><em>tf</em></td>
<td>0.827752</td>
<td>0.884352</td>
</tr>
<tr>
<td><em>tf.idf</em></td>
<td>no</td>
<td></td>
<td></td>
<td>0.626888</td>
<td>0.681446</td>
</tr>
<tr>
<td><em>tf.idf</em></td>
<td>no</td>
<td>10</td>
<td><em>tf.idf</em></td>
<td>0.635875</td>
<td>0.682134</td>
</tr>
<tr>
<td><em>tf.idf</em></td>
<td>yes</td>
<td>10</td>
<td><em>tf.idf</em></td>
<td>0.718177</td>
<td>0.760925</td>
</tr>
<tr>
<td><em>tf.idf</em></td>
<td>yes</td>
<td>20</td>
<td><em>tf.idf</em></td>
<td>0.715399</td>
<td>0.762961</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>no</td>
<td>10</td>
<td>$\chi^2$</td>
<td>0.847393</td>
<td>0.841513</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>no</td>
<td>20</td>
<td>$\chi^2$</td>
<td>0.853436</td>
<td>0.849575</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>yes</td>
<td>10</td>
<td>$\chi^2$</td>
<td>0.822615</td>
<td>0.828412</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>yes</td>
<td>20</td>
<td>$\chi^2$</td>
<td>0.839114</td>
<td>0.839055</td>
</tr>
<tr>
<td><em>With contrast set</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>tf.idf</em></td>
<td>no</td>
<td></td>
<td><em>tf.idf</em></td>
<td>0.626888</td>
<td>0.681446</td>
</tr>
<tr>
<td><em>tf.idf</em></td>
<td>no</td>
<td>10</td>
<td><em>tf.idf</em></td>
<td>0.635875</td>
<td>0.682134</td>
</tr>
<tr>
<td><em>tf.idf</em></td>
<td>yes</td>
<td>10</td>
<td><em>tf.idf</em></td>
<td>0.718177</td>
<td>0.760925</td>
</tr>
<tr>
<td><em>tf.idf</em></td>
<td>yes</td>
<td>20</td>
<td><em>tf.idf</em></td>
<td>0.715399</td>
<td>0.762961</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>no</td>
<td>10</td>
<td>$\chi^2$</td>
<td>0.847393</td>
<td>0.841513</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>no</td>
<td>20</td>
<td>$\chi^2$</td>
<td>0.853436</td>
<td>0.849575</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>yes</td>
<td>10</td>
<td>$\chi^2$</td>
<td>0.822615</td>
<td>0.828412</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>yes</td>
<td>20</td>
<td>$\chi^2$</td>
<td>0.839114</td>
<td>0.839055</td>
</tr>
<tr>
<td><em>Varying partitions</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>yes</td>
<td>30/0</td>
<td>$\chi^2$</td>
<td>0.912525</td>
<td>0.881494</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>yes</td>
<td>30/3</td>
<td>$\chi^2$</td>
<td>0.903534</td>
<td>0.879115</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>yes</td>
<td>30/6</td>
<td>$\chi^2$</td>
<td>0.903611</td>
<td>0.873444</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>yes</td>
<td>30/9</td>
<td>$\chi^2$</td>
<td>0.899407</td>
<td>0.868053</td>
</tr>
</tbody>
</table>
Web signature experiment

Procedure:
1. Create query from Ontology concept (word + defn. words)
2. Retrieve ~5,000 documents (8 web search engines)
3. Purify results (remove duplicates, html, etc.)
4. Extract word family (using \textit{tf.idf}, $\chi^2$, LSA, etc.)
5. Purify
6. Compare to siblings and parents in the Ontology

Problem: raw signatures overlap…
- average parent-child node overlap: \textasciitilde50\%
- Bakery—Edifice: \textasciitilde35\% …too far: missing generalization.
- Airplane—Aircraft: \textasciitilde80\% …too close?

Remaining problem: web signatures still not pure...

WordNet: In 2002–04, Agirre and students (U of the Basque Country) built signatures for all WordNet nouns
Recent work using signatures

• **Multi-document summarization** (Lin and Hovy, 2002)
  – Create \( \lambda \) signature for each set of texts.
  – Create IR query from signature terms; use IR to extract sentences.
  – (Then filter and reorder sentences into single summary).
  – **Performance:** DUC-01: tied first; DUC-02: tied second place

• **Wordsense disambiguation** (Agirre, Ansa, Martinez, Hovy, 2001)
  – Try to use WordNet concepts to collect text sets for signature creation:
    
    \[
    \text{word} + \text{synonym} > \text{def-words} > \text{word .AND. synonym .NEAR. def-word} > \text{etc} \ldots.
    \]
  – Built competing signatures for various noun senses:
    (a) WordNet synonyms; (b) SemCor tagged corpus (\( \chi^2 \));
    (c) web texts (\( \chi^2 \)); (d) WSJ texts (\( \chi^2 \)).
  – **Performance:** Web signatures > random, WordNet baseline.

• **Email clustering** (Murray and Hovy, in prep)
  – Social Network Analysis: Cluster emails and create signatures.
  – Infer personal expertise, project structure, experts omitted, etc.
  – Corpora: ENRON (240K emails), ISI corpus, NSF eRulemaking corpus.
3a. Data Instances:
Extracting seed instances from text

(This work with Michael Fleischman)
What kinds of knowledge?

• **Goal 1**: Add instanical knowledge:
  – Sofia is a city
  – Sofia is a woman’s name
  – Cleopatra was a queen
  – Everest is a mountain
  – Varig is an airline company

• **Goal 2**: Add definitional / descriptive knowledge:
  – Mozart was born in 1756
  – Bell invented the telephone
  – Pisa is in Italy
  – The Leaning Tower is in Pisa
  – Columbus discovered America

• **Uses**:
  – QA (answer suggestion and validation)
  – Wordsense disambiguation for MT

• **Sources**:
  – Existing lists (CIA factbook, atlases, phone books…)
  – Dictionaries and encyclopedias
  – The Web

Create links between concepts
Classify instances under types
Learning about locations… (Fleischman 01)

• Challenge ex.—*region, state/territory, or city?*

  - The company, which is based in *Dpiyj Dsm Gtsmdodvp, Vsaog.*, said an antibody prevented development of paralysis.
  - The situation has been strained ever since *Yplup* began waging war in *Dpiyj Rsdy Sdos*.
  - The pulp and paper operations moved to *Dpiyj Vstpaomos* in 1981.

• Try to learn instances of 8 types (*country, region, territory, city, street, artifact, mountain, water*):
  - (we have lists of these already, so finding sentences for training data is easy).

• Uses:
  - QA: corroborating evidence for answer.
  - IR: query expansion and signature enrichment.

*South San Francisco—region; Calif.—state; Tokyo—city; South East Asia—region; South Carolina—state*
Learning procedure

**Approach:**
- **Training:** For each location, identify features in context; try to learn features that indicate each type
- **Usage:** For new material, use learned features to classify type of location; place results with high confidence into ontology

**Training:**
- Applied BBN’s IdentiFinder to bracket locations
- Chose 88 features (unigrams, bigrams, trigrams in fixed positions before and after location instance; later added signatures, etc.)
- 3 approaches: *Bayesian classifier*, *neural net*, *decision tree* (C4.5)
- MemRun procedure: store good examples and prefer later
Memrun

Initial results:
- Bayesian classifier not very accurate; neural net ok.
- D-tree better, but still multiple classes for each instance.

Memrun: record best example of each instance.

Algorithm with Memrun:
- **Pass 1:** for each text,
  - preprocess with POS tagger and IdentiFinder
  - apply D-tree to classify instance
  - if $score > \text{THRESH1}$, save $(\text{instance}, \text{tag}, \text{score})$ in Memrun
- **Pass 2:** for each text,
  - again apply D-tree
  - if $score < \text{THRESH2}$, replace tag by Memrun value
Examples

City
Aachen
Abadan
Abassi Madani
Abbassi Madani
Abbreviations AZ
Abullojonov
Aberdeen
Abidjan
Abidjan Radio Cote d'Ivoire Chaine
Abiko
Abrahamite
Abramenkov
Abu Dhabi
Abuja
Abyssinia
Acari
Accom
Accordance
...
Results for locations

<table>
<thead>
<tr>
<th>Decision Tree</th>
<th>After C4.5</th>
<th>After MemRun</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>%</td>
<td>%</td>
</tr>
<tr>
<td>City</td>
<td>69.70</td>
<td>82.80</td>
</tr>
<tr>
<td>Country</td>
<td>85.30</td>
<td>91.20</td>
</tr>
<tr>
<td>Street</td>
<td>60.00</td>
<td>60.00</td>
</tr>
<tr>
<td>Region</td>
<td>67.70</td>
<td>67.70</td>
</tr>
<tr>
<td>Water</td>
<td>85.70</td>
<td>85.70</td>
</tr>
<tr>
<td>Artifacts</td>
<td>50.00</td>
<td>50.00</td>
</tr>
<tr>
<td>Territory</td>
<td>48.00</td>
<td>53.40</td>
</tr>
<tr>
<td>Mount</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Total</td>
<td>73.20</td>
<td>81.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Confusion Matrix</th>
<th>After C4.5</th>
<th>After MemRun</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td>CITY</td>
<td>COUNTR</td>
</tr>
<tr>
<td>CITY</td>
<td>309</td>
<td>51</td>
</tr>
<tr>
<td>COUNTRY</td>
<td>29</td>
<td>440</td>
</tr>
<tr>
<td>STREET</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>REGION</td>
<td>12</td>
<td>7</td>
</tr>
<tr>
<td>WATER</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ARTS</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>TERR</td>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td>MOUNT</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Human Competence</th>
<th>Subject 1</th>
<th>Subject 2</th>
<th>Subject 3</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>%</td>
<td>%</td>
<td>%</td>
<td>%</td>
</tr>
<tr>
<td>City</td>
<td>71.4</td>
<td>71.4</td>
<td>57.1</td>
<td>66.7</td>
</tr>
<tr>
<td>Country</td>
<td>73</td>
<td>67.6</td>
<td>45.9</td>
<td>62.2</td>
</tr>
<tr>
<td>Street</td>
<td>40</td>
<td>40</td>
<td>20</td>
<td>33.3</td>
</tr>
<tr>
<td>Region</td>
<td>50</td>
<td>50</td>
<td>37.5</td>
<td>45.8</td>
</tr>
<tr>
<td>Water</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Artifacts</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>66.7</td>
</tr>
<tr>
<td>Territory</td>
<td>72.7</td>
<td>45.5</td>
<td>90.9</td>
<td>69.7</td>
</tr>
<tr>
<td>Mount</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>65.3</td>
<td>53.3</td>
<td>53.3</td>
<td>57.3</td>
</tr>
</tbody>
</table>

THRESHOLDS vs. YIELD Decision Tree

THRESH1 = 77%; THRESH2 = 98%

NB: test samples are small
People

(Fleischman and Hovy 02)

Goal: Collecting training data about 8 types of people: politicians, entertainers (movie stars, etc.), athletes, businesspeople...

Procedure: as before, with added features using signature of each category and WordNet hypernyms.

| athlete  | 458.029 | ;       | 398.626 | perez  | 392.904 | rogers  | 368.441 | carlos  | 351.686 | points  | 333.083 | roy      | 311.042 | andres  | 284.927 | scored  | 273.197 | chris   | 252.172 | hardee's | 239.747 | george  | 223.879 | games   | 222.202 | mark    | 217.711 | mike    | ...
| businessperson | 4428.267 | greenspan | 3999.135 | alan    | 2774.682 | reserve | 2429.129 | chairman | 1786.783 | federal | 1709.120 | icahn | 1665.358 | fed     | 1252.701 | carl    | 827.029 | board   | 682.420 | rates   | 662.510 | investor | 651.529 | twa     | 531.907 | kerkorian | 522.072 | interest | ...
| cleric          | 1133.793 | rabbi    | 1074.785 | cardinal | 1011.190 | paul    | 809.128 | archbishop | 798.372 | john    | 748.170 | bishop | 714.173 | catholic | 688.291 | church | 613.625 | roman | 610.287 | tutu | 584.720 | desmond | 460.057 | pope | 309.923 | kahane | 300.236 | meir | 404.008 | said | ...
| entertainer     | 1902.178 | "        | 1573.695 | actor   | 1083.622 | actress | 721.929 | movie | 618.947 | george | 607.466 | film     | 553.659 | singer | 541.235 | president | 536.962 | her | 536.856 | keating | 528.226 | star | 448.524 | ( | 433.065 | ) | 404.008 | said |...
### Best results:

Using signatures and WordNet hyperlinks (but no synset expansion).

### Problems:

- Training and test data skewed.
- Genuine ambiguity often, politician = military leader.

#### Confusion Matrix

<table>
<thead>
<tr>
<th>Type</th>
<th>BUS</th>
<th>POL</th>
<th>ENT</th>
<th>MIL</th>
<th>DOC</th>
<th>ATH</th>
<th>CLR</th>
<th>LAW</th>
<th>COP</th>
<th>MISC</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS</td>
<td>120</td>
<td>23</td>
<td>7</td>
<td>1</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>POL</td>
<td>8</td>
<td>461</td>
<td>34</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ENT</td>
<td>12</td>
<td>6</td>
<td>150</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MIL</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DOC</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>48</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ATH</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CLR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LAW</td>
<td>0</td>
<td>29</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COP</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>MISC</td>
<td>9</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>
Instance extraction++ (Fleischman & Hovy 03)

• **Goal**: extract *all* instances from the web

• **Method**:
  – Download text from web (15GB)
  – Identify named entities (BBN’s IdentiFinder (Bikel et al. 93))
  – Extract ones with descriptive phrases (\(<\text{APOS}>, <\text{CN/PN}>\))
    (“the vacuum manufacturer Horeck” / “Saddam’s physician Abdul”)
  – Cluster them, and categorize in ontology

• **Result**: over 900,000 instances
  – Average: 2 mentions per instance, 40+ for George W. Bush

• **Evaluation**:
  – Tested with 200 “who is X?” questions
  – Better than TextMap: 25% more
  – Faster: 10 sec ↔ 9 hr!

Performance on a Question Answering Task

<table>
<thead>
<tr>
<th>% Correct</th>
<th>State of the Art System</th>
<th>Extraction System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partial</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>Correct</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td>Incorrect</td>
<td>70</td>
<td>50</td>
</tr>
</tbody>
</table>
3b. Learning Relations: Harvesting additional information from text

(This work with Deepak Ravichandran and Patrick Pantel)
Shallow patterns for information

- **Goal**: learn relationship data from the web
  - (when was someone born? Where does he live?)

- **Procedure**: automatically learn **word-level patterns**
  
  When was Mozart born?
  “Mozart (1756–1792)…”

  \[
  \text{[ } \text{<NAME> ( <BIRTHYEAR> – <DEATHYEAR> ) } \text{]} \]

- **Apply patterns to Omega concepts/instances**

- **Evaluation**: test in TREC QA competition

- **Main problem**: learning patterns
  
  (In TREC QA 2001, Soubbotin and Soubbotin got very high score with over 10,000 patterns built by hand)
Learning extraction patterns from the web

(Ravichandran and Hovy 02)

• Prepare:
  – Select example for target relation: Q term (Mozart) and A term (1756)

• Collect data:
  – Submit Q and A terms as queries to a search engine (Altavista)
  – Download top 1000 web documents

• Preprocess:
  – Apply a sentence breaker to the documents
  – Retain only sentences with both Q and A terms
  – Pass retained sentences through suffix tree constructor

• Select and create patterns:
  – Filter each phrase in the suffix tree to retain only those phrases that contain both Q and A terms
  – Replace the Q term by the tag “<NAME>” and the A term by the term by “<ANSWER>”
Some results

BIRTHYEAR:
1.0  <NAME> (<ANS> —
0.85  <NAME> was born on <ANS>
0.6  <NAME> was born in <ANS>
...

DEFINITION:
1.0  <NAME> and related <ANS>s
1.0  <ANS> (<NAME>,
0.9  as <NAME> , <ANS> and
...

LOCATION:
1.0  <ANS>’s <NAME>.
1.0  regional : <ANS> : <NAME>
0.9  the <NAME> in <ANS> ,
...

<table>
<thead>
<tr>
<th>Testing (TREC-10 questions)</th>
<th>Question type</th>
<th>Num Qs</th>
<th>TREC MRR</th>
<th>Web MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIRTHYEAR</td>
<td>8</td>
<td>0.479</td>
<td>0.688</td>
<td></td>
</tr>
<tr>
<td>INVENTOR</td>
<td>6</td>
<td>0.167</td>
<td>0.583</td>
<td></td>
</tr>
<tr>
<td>DISCOVERER</td>
<td>4</td>
<td>0.125</td>
<td>0.875</td>
<td></td>
</tr>
<tr>
<td>DEFINITION</td>
<td>102</td>
<td>0.345</td>
<td>0.386</td>
<td></td>
</tr>
<tr>
<td>WHY-FAMOUS</td>
<td>3</td>
<td>0.667</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>LOCATION</td>
<td>16</td>
<td>0.75</td>
<td>0.864</td>
<td></td>
</tr>
</tbody>
</table>
Regular expressions  
(Ravichandran et al. 2004)

- New process: learn regular expression patterns

| Surface | Babe Ruth was born in Baltimore, on February 6, 1895 |
| NE Tags  | <NAME> <LOCATION> <DATE> |
| Part of Speech | NNP NNP VBD VBN IN NNP, IN NNP CD |

| Surface | George Herman "Babe" Ruth was born here in 1895 |
| NE Tags  | <NAME> |
| Part of Speech | NNP NNP NNP NNP VBD VBN RB IN CD |

- Complexity: $O(y^2)$, for max string length $y$
- Results: over 2 million instances from 15GB corpus
Comparing clustering and surface patterns

- Precision: took random 50 words, each with system’s learned superconcepts (top 3 of system); added top 3 from WordNet, 1 human superconcept. Used 2 judges (Kappa = 0.78–0.85)
- Recall: \[ \text{Relative Recall} = \frac{\text{Recall}_{\text{patt}}}{\text{Recall}_{\text{co-occ}}} = \frac{C_P}{C_C} \]
- TREC-03 def’ns: Patt up to 52%; Co-Occ up to 44% MRR

Relative Recall

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Pattern System</th>
<th>Co-occurrence System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec</td>
<td>Top-3</td>
</tr>
<tr>
<td>1.5MB</td>
<td>56.6%</td>
<td>60.0%</td>
</tr>
<tr>
<td>15MB</td>
<td>57.3%</td>
<td>63.0%</td>
</tr>
<tr>
<td>150MB</td>
<td>50.7%</td>
<td>56.0%</td>
</tr>
<tr>
<td>1.5GB</td>
<td>52.6%</td>
<td>51.0%</td>
</tr>
<tr>
<td>15GB</td>
<td>61.8%</td>
<td>69.0%</td>
</tr>
<tr>
<td>150GB</td>
<td>67.8%</td>
<td>67.0%</td>
</tr>
</tbody>
</table>
4a. Verifying Data: Determining opinion (and other epistemic statuses, one day)

(This work with Soo-Min Kim)
Summary of Opinion detection (Kim and Hovy, 2004)

• Our definition of Opinion:
  – A quadruple [Topic, Holder, Claim, Sentiment]
  – Holder = person or organization
  – Claim = statement about Topic (“abortion should be banned”)
  – Sentiment (only with affective opinions):
    • Positive (Claim is GOOD) or Negative or Mixed or Neutral (“I don’t care one way or the other”) or Unstated (“they had strong feelings about that”)

• Approach:
  1. Opinion recognition: Find opinion-bearing expressions in texts
  2. Sentiment recognition: For affective opinions, determine the Sentiment of these expressions

• Algorithms:
  – Word sentiment classification (for individual verbs, adjectives)
  – Sentence sentiment classification (tested various models, for different combinations of word sentiment classifiers)
Evaluation results (accuracy)

- Human-human agreement: 88.9% (adjs) and 85.1% (verbs)
- Human-system agreement: 76.8% (adjs) and 80.1% (verbs), with recall 97.8% and 93.2%

\[ \text{mi: sentence classifier model} \]
\[ \text{pi: word classifier model} \]

<table>
<thead>
<tr>
<th></th>
<th>Full sent</th>
<th>Holder–Topic</th>
<th>H–T ±2 words</th>
<th>H/T to sent end</th>
</tr>
</thead>
<tbody>
<tr>
<td>m0p1</td>
<td>0.772727</td>
<td>0.742424</td>
<td>0.712121</td>
<td>0.80303</td>
</tr>
<tr>
<td>m0p3</td>
<td>0.666667</td>
<td>0.727273</td>
<td>0.712121</td>
<td>0.818182</td>
</tr>
<tr>
<td>m1p1</td>
<td>0.651515</td>
<td>0.712121</td>
<td>0.69697</td>
<td>0.712121</td>
</tr>
<tr>
<td>m1p2</td>
<td>0.69697</td>
<td>0.742424</td>
<td>0.742424</td>
<td>0.772727</td>
</tr>
<tr>
<td>m1p3</td>
<td>0.787879</td>
<td>0.757576</td>
<td>0.727273</td>
<td>0.757576</td>
</tr>
<tr>
<td>m1p4</td>
<td>0.681818</td>
<td>0.772727</td>
<td>0.787879</td>
<td>0.787879</td>
</tr>
<tr>
<td>m2p1</td>
<td>0.69697</td>
<td>0.727273</td>
<td>0.727273</td>
<td>0.787879</td>
</tr>
<tr>
<td>m2p2</td>
<td>0.590909</td>
<td>0.681818</td>
<td>0.651515</td>
<td>0.681818</td>
</tr>
<tr>
<td>m2p3</td>
<td>0.636364</td>
<td>0.742424</td>
<td>0.742424</td>
<td>0.742424</td>
</tr>
<tr>
<td>m2p4</td>
<td>0.590909</td>
<td>0.69697</td>
<td>0.666667</td>
<td>0.681818</td>
</tr>
</tbody>
</table>
5. Conclusion
Summary

Ingredients:
- small ontologies and metadata sets
- concept families (signatures)
- information from dictionaries, etc.
- additional info from text and the web

Method:
1. Into a large database, pour all ingredients
2. Stir together in the right way
3. Bake

Evaluate — IR, QA, MT, and so on!
What would be nice?

• Databases that support rapid reorganization when new metadata is learned
• Databases that accommodate possibly inconsistent, partial, and tenuous data
• Very very large databases that can grow rapidly
Current status

• Omega:
  – Approx. 110,000 concepts
  – Approx. 1.1 mill instances
  – Subject information from TAP (Guha et al.)
  – Additional information from various sources

• Tools:
  – Alignment algorithms
  – Concept spotting, clustering, glossary parsing algorithms
  – Instance harvesting algorithm
  – Algorithms to learn inter-concept/instance relations

• Infrastructure:
  – Instances (and concepts?) into RDF (also database form?)
  – Online access and DINO browser
Next steps

• Collect **instances** of many other entities (not only locations, organizations, and people)
• Learn **more details** about each person, location, organization, etc., using patterns: date of birth, nationality, occupation, spouse, etc.
• Into Omega, incorporate **WordNet extensions** (inference rules) from (Moldovan 03)
• Merge **OpenCyc** and perhaps SIC code terms into Omega
• Build **access tools** and inline access methods to support QA, summarization, etc.
Vision

• Many people could use something like Omega:
  – The Semantic Web needs a large standardized well-organized multi-lingual termset
  – MT systems need a language-independent (or at least neutral) ontology
  – Many HLT systems can use the semantic and instantial information in Omega for better performance
  – Database integration and access systems might use something like Omega
  – AI systems might take subsets of it

• People should be encouraged to build their own!
Thank you